DOCUMENTATION

Assignment 2

QUEUE MANAGEMENT SYSTEM USING THREADS AND SYNCHRONIZATION MECHANISMS

Student: Nurtayeva Dyuldirim
Grupa: 7

Contents

1. Assignment Objective.....3
2. Problem Analysis, Modeling, Scenarios, Use Cases.....3
3. Design.....4
4. Implementation.....6-11
5. Conclusion.....12
6. Bibliography.....12

1. Assignment Objective

The implementation of a specific application for simulating queues, which focuses on minimizing customer waiting time. The application takes input from a specified file containing customer data, and then simulates the distribution of customers and their waiting times.

The main goal of the application is to provide a waiting space for customers to minimize their wait time. The application can be run in either a graphical or a physical waiting area, depending on the context of the application.

The application aims to minimize the waiting time for customers, which is a common objective in many real-world scenarios. The application simulates scenarios and provides the application with valuable insights into queue management and customer satisfaction.

Overall, it seems that the application is designed to provide a realistic and interactive simulation of queuing systems, with a focus on minimizing customer waiting time and improving the overall customer experience in scenarios such as customer service centers, hospitals, and transportation systems.

2. Problem Analysis, Modeling, Scenarios, Use Cases

The simulation model for a queuing system consists of a set of queues, and customers are assigned to the queue with the shortest waiting time. The objective is to evaluate the performance of the system in terms of waiting times and queue lengths under different scenarios.

The simulation model has two policies for adding customers to waiting queues. The first policy is to add a customer to the shortest queue, and the second policy is to add a customer to a queue randomly if the shortest queue is full. This allows the simulation to test different scenarios and compare the performance of the system.

The simulation model must display the status of the queues every second. The status should include information about the number of customers in each queue and regular intervals. The status of the queues can be either open, which means that there are customers waiting in the queue, or closed, which means that the queue is empty and no customers are waiting.

The simulation model also includes a priority system for processing customers. This means that when a priority customer arrives, they are processed before other customers in the queue. If multiple priority customers arrive, they are removed from the queue, and the next customer can be processed.

Overall, it seems that the simulation model is designed to evaluate the performance of a queuing system under different scenarios and policies. The simulation model can be useful for optimizing the system and improving customer service.

The operation of adding a customer to the queue	
Brief description	Based on the strategy chosen by the simulator, a customer is added in a queue.
Actors	Doesn't need an actor.
Priority	has priority
Trigger	When the client's waiting time is equal to the time when the customer is ready to be adding in a queue is achieved.
Preconditions	Have positive waiting times and positive service times so that they can be added in a queue and be served.
Postconditions	Have zero service times so that they are removed from the queue.
Expected run	1. The customer is on the waiting list. 2. The arrival time is identical to the simulation time. 3. Choose the most favorable queue based on the total number of 4. Add the customer to that queue 5. The customer waits in the queue until picked up. 6. When it is at the top of the queue, the customer is picked up. 7. The customer's service time is reduced every second 8. When the customer service time reaches zero, it is completed and subsequently removed the customer from the queue

3. Design

The system is a simulation program that generates and writes data about customers and processes them using a queuing system. The program includes files that implement the simulator , write the GUI package deals with the graphical interface, the data package stores the data files for successful simulation runs.

This organization is sensible as it separates the concerns of different components into separate packages, making it easier to maintain and modify each component without affecting the other components. The graphical interface, The Business Logic package is responsible for the simulation, data processing, and other core functionalities of the system. The separation of responsibilities is a good practice for building maintainable and scalable software.

src
data
gui
.gitignore
.idea
.DS_Store
.gitignore
.idea
.DS_Store
.gitignore
.idea
.DS_Store